In [48]:

```
import pandas as pd
import warnings
warnings.filterwarnings('ignore')

df = pd.read_csv("DATA/train.csv")
df.head()
```

Out[48]:

	ID	var3	var15	imp_ent_var16_ult1	imp_op_var39_comer_ult1	imp_op_var39_comer_ult3	imp
0	1	2	23	0.0	0.0	0.0	
1	3	2	34	0.0	0.0	0.0	
2	4	2	23	0.0	0.0	0.0	
3	8	2	37	0.0	195.0	195.0	
4	10	2	39	0.0	0.0	0.0	

5 rows × 371 columns

Variance Thresholding

In [2]:

```
from sklearn.feature_selection import VarianceThreshold
vart=VarianceThreshold(threshold=0)
vart.fit(df)
```

Out[2]:

```
VarianceThreshold(threshold=0)
```

In [3]:
```
vart.get_support()
```

Out[3]:
```
array([ True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True, False, False,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True, False, False, False, False,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
       False,  True,  True,  True, False, False,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True, False, False, False,
       False,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True, False,  True,  True,  True,  True,  True,
       False, False,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
       False, False,  True,  True,  True,  True,  True,  True,  True,
        True, False,  True,  True, False,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True, False,  True, False,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True, False,  True,  True,  True, False,  True,  True,  True,
        True,  True, False,  True,  True,  True, False,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True, False, False,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True, False,  True,
        True,  True, False,  True,  True,  True,  True,  True,  True,
        True, False,  True,  True,  True, False,  True,  True,  True,
        True,  True,  True,  True, False,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True, False,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True])
```

In [4]:
```
df.columns[vart.get_support()]
```

Out[4]:
```
Index(['ID', 'var3', 'var15', 'imp_ent_var16_ult1', 'imp_op_var39_comer_ult
1',
       'imp_op_var39_comer_ult3', 'imp_op_var40_comer_ult1',
       'imp_op_var40_comer_ult3', 'imp_op_var40_efect_ult1',
       'imp_op_var40_efect_ult3',
       ...
       'saldo_medio_var33_hace2', 'saldo_medio_var33_hace3',
       'saldo_medio_var33_ult1', 'saldo_medio_var33_ult3',
       'saldo_medio_var44_hace2', 'saldo_medio_var44_hace3',
       'saldo_medio_var44_ult1', 'saldo_medio_var44_ult3', 'var38', 'TARGE
T'],
      dtype='object', length=337)
```

Feature selection using correlation

In [5]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

In [6]:
```
x=df.iloc[:,1:]
y=df.iloc[:,0]
```

In [7]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=
x_train.shape, x_test.shape
```

Out[7]:
```
((53214, 370), (22806, 370))
```

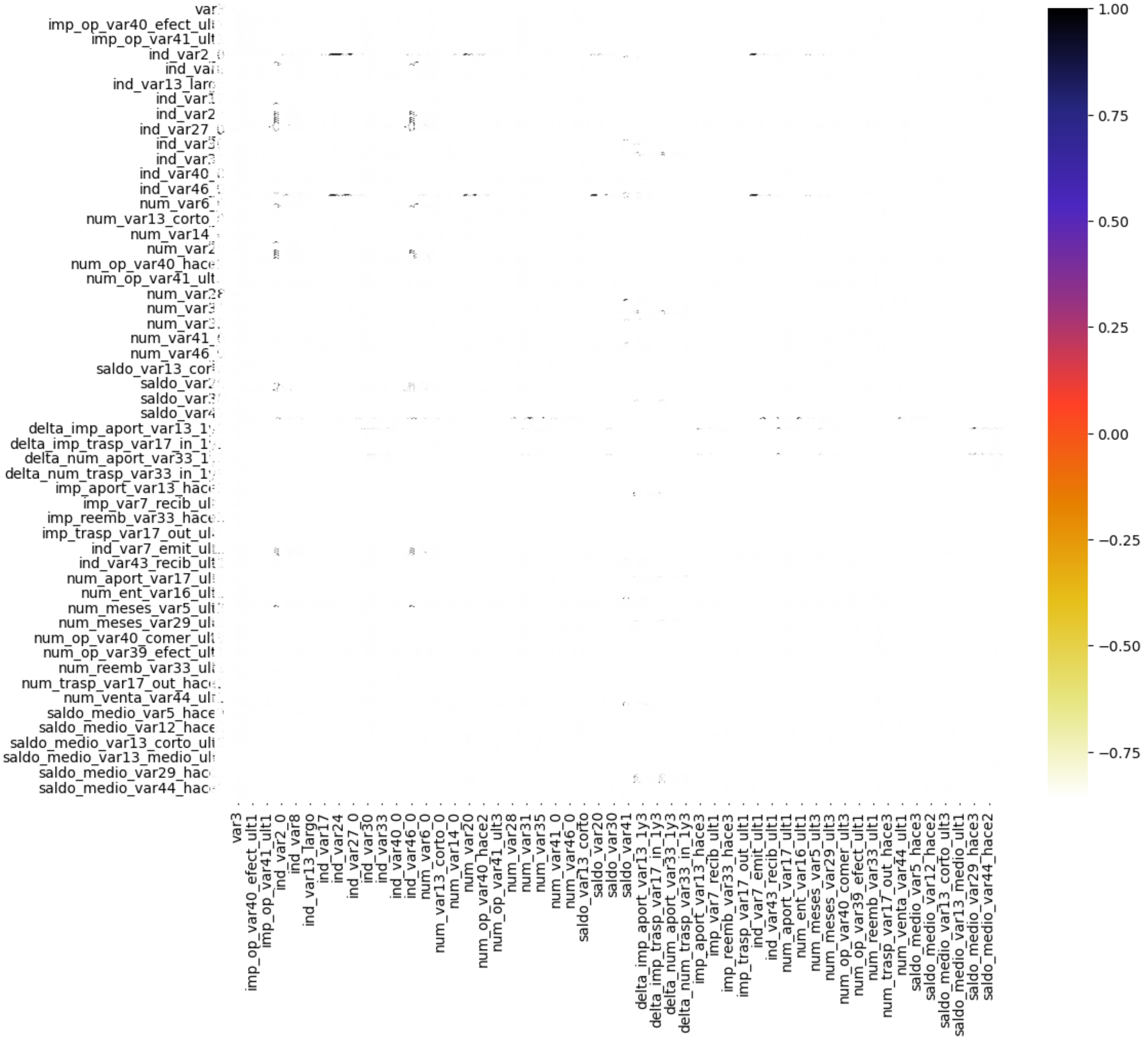
In [8]:
```
x_train.corr()
```

Out[8]:

	var3	var15	imp_ent_var16_ult1	imp_op_var39_comer_ult1	in
var3	1.000000	-0.003769	0.001790	0.005309	
var15	-0.003769	1.000000	0.042432	0.095696	
imp_ent_var16_ult1	0.001790	0.042432	1.000000	0.042590	
imp_op_var39_comer_ult1	0.005309	0.095696	0.042590	1.000000	
imp_op_var39_comer_ult3	0.006322	0.101386	0.035779	0.889181	
...	...	...	...	...	
saldo_medio_var44_hace3	0.000465	0.019212	-0.000595	0.008638	
saldo_medio_var44_ult1	0.000769	0.034742	0.006117	0.013411	
saldo_medio_var44_ult3	0.000805	0.034995	0.008007	0.012702	
var38	0.000140	0.004416	-0.000348	0.009369	
TARGET	0.005672	0.099938	-0.001677	0.008450	

370 rows × 370 columns

```
In [10]: import seaborn as sns
         plt.figure(figsize=(12,10))
         cor=x_train.corr()
         sns.heatmap(cor,annot=True,cmap=plt.cm.CMRmap_r)
         plt.show()
```



```
In [11]: def correlation(dataset, threshold):
             col_corr = set()  # Set of all the names of correlated columns
             corr_matrix = dataset.corr()
             for i in range(len(corr_matrix.columns)):
                 for j in range(i):
                     if abs(corr_matrix.iloc[i, j]) > threshold: # we are interested in
                         colname = corr_matrix.columns[i]  # getting the name of column
                         col_corr.add(colname)
             return col_corr
```

```
In [12]: corr_f=correlation(x_train,0.7)
         len(set(corr_f))
```

```
Out[12]: 232
```

In [13]:
```
corr_f
```

Out[13]:
```
{'delta_imp_amort_var18_1y3',
 'delta_imp_amort_var34_1y3',
 'delta_num_aport_var13_1y3',
 'delta_num_aport_var17_1y3',
 'delta_num_aport_var33_1y3',
 'delta_num_compra_var44_1y3',
 'delta_num_reemb_var13_1y3',
 'delta_num_reemb_var17_1y3',
 'delta_num_trasp_var17_in_1y3',
 'delta_num_trasp_var17_out_1y3',
 'delta_num_trasp_var33_in_1y3',
 'delta_num_trasp_var33_out_1y3',
 'delta_num_venta_var44_1y3',
 'imp_amort_var18_ult1',
 'imp_amort_var34_ult1',
 'imp_aport_var13_ult1',
 'imp_aport_var33_hace3',
 'imp_op_var39_comer_ult3',
 'imp_op_var39_efect_ult1',
```

In [14]:
```
x_train.drop(corr_f,axis=1)
x_test.drop(corr_f,axis=1)
```

Out[14]:

	var3	var15	imp_ent_var16_ult1	imp_op_var39_comer_ult1	imp_op_var40_comer_ult1	im
36443	2	23	0.0	0.00	0.0	
17577	2	23	0.0	0.00	0.0	
28097	2	44	0.0	0.00	0.0	
34967	2	24	0.0	0.00	0.0	
50314	2	80	0.0	0.00	0.0	
...	...	...	...	...	...	
49159	2	40	0.0	1086.00	0.0	
38812	2	43	0.0	552.30	0.0	
60399	2	23	0.0	0.00	0.0	
40783	2	31	0.0	426.18	0.0	
73639	2	23	0.0	0.00	0.0	

22806 rows × 138 columns

Chi-Square Test

In [62]:
```
import pandas as pd
df2=pd.read_csv("Data/ml book2.csv")
df2
```

Out[62]:

	Day	Outlook	temp	humidity	windy	play
0	1	sunny	hot	high	False	NO
1	2	sunny	hot	high	True	NO
2	3	Overcast	hot	high	False	YES
3	4	rainy	mild	high	False	YES
4	5	rainy	cold	Normal	False	YES
5	6	rainy	cold	Normal	True	NO
6	7	overcast	cold	Normal	True	YES
7	8	sunny	mild	high	False	NO
8	9	sunny	cold	normal	False	YES
9	10	rainy	mild	normal	False	YES
10	11	sunny	mild	normal	True	YES
11	12	overcast	mild	high	True	YES
12	13	overcast	hot	normal	False	YES
13	14	rainy	mild	high	True	NO

In [63]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14 entries, 0 to 13
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Day       14 non-null     int64 
 1   Outlook   14 non-null     object
 2   temp      14 non-null     object
 3   humidity  14 non-null     object
 4   windy     14 non-null     bool  
 5   play      14 non-null     object
dtypes: bool(1), int64(1), object(4)
memory usage: 702.0+ bytes
```

In [64]:
```
##['Day','Outlook','temp','humidity','windy','play']
df2=df2[['Day','Outlook','temp','humidity','windy','play']]
df2.head()
```

Out[64]:

	Day	Outlook	temp	humidity	windy	play
0	1	sunny	hot	high	False	NO
1	2	sunny	hot	high	True	NO
2	3	Overcast	hot	high	False	YES
3	4	rainy	mild	high	False	YES
4	5	rainy	cold	Normal	False	YES

In [65]:
```
from sklearn.preprocessing import LabelEncoder
```

In [66]:
```
le = LabelEncoder()
```

In [67]:
```
df2['Outlook'] = le.fit_transform(df2['Outlook'])
df2['temp'] = le.fit_transform(df2['temp'])
df2['humidity']=le.fit_transform(df2['humidity'])
df2['windy'] = le.fit_transform(df2['windy'])
df2['play'] = le.fit_transform(df2['play'])
```

In [68]:
```
df2
```

Out[68]:

	Day	Outlook	temp	humidity	windy	play
0	1	3	1	1	0	0
1	2	3	1	1	1	0
2	3	0	1	1	0	1
3	4	2	2	1	0	1
4	5	2	0	0	0	1
5	6	2	0	0	1	0
6	7	1	0	0	1	1
7	8	3	2	1	0	0
8	9	3	0	2	0	1
9	10	2	2	2	0	1
10	11	3	2	2	1	1
11	12	1	2	1	1	1
12	13	1	1	2	0	1
13	14	2	2	1	1	0

In [69]:

```
x = df2.iloc[:,:-1]
y = df2.iloc[:,-1]
```

In [70]:

```
## Perform chi2 test
### chi2 returns 2 values
### Fscore and the pvalue
from sklearn.feature_selection import chi2
f_p_values=chi2(x,y)
```

In [71]:

```
f_p_values
```

Out[71]:

```
(array([1.75259259, 1.4       , 0.02222222, 0.53481481, 0.53333333]),
 array([0.18555114, 0.23672357, 0.88149745, 0.46458962, 0.46520882]))
```

In [72]:

```
import pandas as pd
p_values=pd.Series(f_p_values[0])
p_values.index=x.columns
p_values
```

Out[72]:

```
Day         1.752593
Outlook     1.400000
temp        0.022222
humidity    0.534815
windy       0.533333
dtype: float64
```

In [73]:

```
p_values.sort_index(ascending=False)
```

Out[73]:

```
windy       0.533333
temp        0.022222
humidity    0.534815
Outlook     1.400000
Day         1.752593
dtype: float64
```